A picture containing logo

Description automatically generated

**AMMY Pharmacy**

**CSCI313: Software Engineering Project**

**Software Requirements**

**Specifications**

UNDER THE SUPERVISION OF:

**Dr. Ahmed Fathy Elnokrashy**

**Eng. ElShaymaa Sayed ElKaliouby**

MADE BY:

|  |  |
| --- | --- |
| **Mostafa Samer Dorrah** | **202000125** |
| **Mohamed Ahmed Mohamed** | **202000991** |
| **Yousif Adel Khalil** | **202000238** |
| **Abdelrahman Mohamed Elatrozy** | **202001188** |
|  |  |

# Table of content

[Table of content 2](#_Toc122801339)

[**1-** **Introduction** 5](#_Toc122801340)

[**1.1** **Purpose** 5](#_Toc122801341)

[**1.2** **SCOPE** 5](#_Toc122801342)

[**1.3** **TECHNOLOGIES USED** 5](#_Toc122801343)

[**1.4** **Intended audience** 6](#_Toc122801344)

[**2-** **Overall Description** 6](#_Toc122801345)

[**2.1 Product Perspective** 6](#_Toc122801346)

[**2.2 User Characteristics** 6](#_Toc122801347)

[**2.3 Operating environment** 7](#_Toc122801348)

[**2.4 Constraints** 7](#_Toc122801349)

[**2.4.1 Software constraints** 7](#_Toc122801350)

[**2.5 Assumptions & dependencies** 7](#_Toc122801351)

[**3-** **Interfaces** 7](#_Toc122801352)

[**3.1 System Interface** 7](#_Toc122801353)

[**4-Functional Requirements** 9](#_Toc122801354)

[4.1 Customer class 9](#_Toc122801355)

[4.1.1 Functional Requirements 9](#_Toc122801356)

[4.1.2 Functional Requirements 10](#_Toc122801357)

[4.1.3 Functional Requirements 10](#_Toc122801358)

[4.1.4 Functional Requirements 10](#_Toc122801359)

[4.1.5 Functional Requirements 10](#_Toc122801360)

[4.1.6 Functional Requirements 11](#_Toc122801361)

[4.1.7 Functional Requirements 11](#_Toc122801362)

[4.1.8 Functional Requirements 11](#_Toc122801363)

[4.1.9 Functional Requirements 11](#_Toc122801364)

[4.1.10 Functional Requirements 11](#_Toc122801365)

[4.2 aDMIN class 12](#_Toc122801366)

[4.2.1 Functional Requirements 12](#_Toc122801367)

[4.2.2 Functional Requirements 12](#_Toc122801368)

[4.2.3 Functional Requirements 12](#_Toc122801369)

[4.2.4 Functional Requirements 13](#_Toc122801370)

[4.2.5 Functional Requirements 13](#_Toc122801371)

[4.3 aDMIN class 13](#_Toc122801372)

[4.3.1 Functional Requirements 13](#_Toc122801373)

[4.3.2 Functional Requirements 13](#_Toc122801374)

[4.3.3 Functional Requirements 14](#_Toc122801375)

[4.3.4 Functional Requirements 14](#_Toc122801376)

[4.4 Cart class 14](#_Toc122801377)

[4.4.1 Functional Requirements 14](#_Toc122801378)

[4.4.2 Functional Requirements 14](#_Toc122801379)

[4.4.3 Functional Requirements 14](#_Toc122801380)

[**5-Non-Functional requirement** 15](#_Toc122801381)

[5.1 Reliability 15](#_Toc122801382)

[5.2 Recoverability 15](#_Toc122801383)

[5.3 Performance 15](#_Toc122801384)

[5.4 Availability 15](#_Toc122801385)

[5.5 Usability 15](#_Toc122801386)

[5.6 Maintainability 15](#_Toc122801387)

[5.7 Security 15](#_Toc122801388)

[**6-Diagrams** 16](#_Toc122801389)

[6.1 Use case diagram 16](#_Toc122801390)

[6.1.1 Use case scenarios 16](#_Toc122801391)

[6.1.2 Use case scenarios 17](#_Toc122801392)

[6.1.3 Use case scenarios 18](#_Toc122801393)

[6.1.4 Use case scenarios 19](#_Toc122801394)

[6.1.5 Use case scenarios 20](#_Toc122801395)

[6.1.6 Use case scenarios 21](#_Toc122801396)

[6.1.7 Use case scenarios 22](#_Toc122801397)

[6.1.8 Use case scenarios 23](#_Toc122801398)

[6.1.9 Use case scenarios 24](#_Toc122801399)

[6.1.10 Use case scenarios 25](#_Toc122801400)

[6.1.11 Use case scenarios 26](#_Toc122801401)

[6.1.12 Use case scenarios 27](#_Toc122801402)

[6.2-Sequence diagrams 28](#_Toc122801403)

[6.2.1-ADD Visa 29](#_Toc122801404)

[6.2.2-Confirm Order 29](#_Toc122801405)

[7.3Class Diagram 30](#_Toc122801406)

[8.4 ER DIGRAM 31](#_Toc122801407)

[8.5-Relational Schema 32](#_Toc122801408)

[8.5.1-Business Rules 33](#_Toc122801409)

[9-Test Cases 34](#_Toc122801410)

[9.1-Test Case 01 34](#_Toc122801411)

[9.2-Test Case02 35](#_Toc122801412)

[9.3-Test Case03 35](#_Toc122801413)

[9.4-Test Case04 36](#_Toc122801414)

[9.2-Test Case05 37](#_Toc122801415)

# **Introduction**

## **Purpose**

This document's main objective is to provide a complete explanation of the web application system. It will describe the structure and characteristics of the system, its interfaces, what operations the system will perform, the limitations that must be met for it to operate effectively, and how the system will respond to outside events.

## **SCOPE**

“AMMY Pharmacy” is an online pharmacy system that allow customers and patients to view the products and medicine available in the pharmacy and request order. The system consists of 2 main Users, First the Admins, and the Customers. The admins will have access to the customers and products database. The customers after registration will be able to view the search or any product or medicine, get recommendations, and make purchases.

The application will provide search by active ingredient of the medicine to help our customers to get what they need with minimum cost

## **TECHNOLOGIES USED**

* Python to develop the entire program and system
* MySQL for database
* FastAPI
* ReactJS

## **Intended audience**

This document is useful to: -

**Customer/Pharmacist/Doctor:** To follow up with the project team if there are requirements change.

**Developer:** To make implementation simple.

**Software Tester:** To be able to create the proper test cases for the system and test it to raise the system's level of quality.

**SCRUM Master:** To be able to create a team and work on the project with a strong plan.

# **Overall Description**

## **2.1 Product Perspective**

* “AMMY Pharmacy” application is a self-contained and an independent system with suitable interface.
* The system has two categories of users. 1st category is the user, and the 2nd is the admin. Every category has its own features and functionalities that they can do with the system.

## **2.2 User Characteristics**

There are essentially two types of users can interact with the system, user of web application and administrator; they have a different view of the system.

* The user could be a customer which he/she purchases products that do not require a certain official prescription from a doctor. User can use the system to purchase our products and fulfill his/her needs by either searching for the medicine name or the active ingredient and the system will list all the medicines with the active ingredient the user has typed. This user will only interact with the web application.
* The administrator could be a developer, the admin could interact with the web application, will interact with the database and back end, and will manage the system by managing the users, and by managing medicine data.

## **2.3 Operating environment**

The system will run any computer/laptop with Windows OS, and it is expected to be available also on Linux OS. In addition, the application will be available on any web browser. The application is developed with Python, ReactJS, FastAPI. So, we use the same OS for both development and deployment.

* For development & deployment we use windows OS.

## **2.4 Constraints**

* All type of users shall be connected to the database locally.

### **2.4.1 Software constraints**

* Languages used are Python and a library in JavaScript ‘ReactJS’ for interface.
* The system must use the current standard MySQL database.

## **2.5 Assumptions & dependencies**

* It is assumed that all type of users involved in the use of the application have local connection to the database.

# **Interfaces**

## **3.1 System Interface**

A first-time user of web application should see the log-in page when the user opens the application; go to Figure 01 & Figure 02. If the user is not registered in the system and does not exist in the database system, he/she will be able to open the Sign-Up page. So, the user will create account to be able to use the application.

![Graphical user interface, application

Description automatically generated]()

Figure 01 – Log-in Page

![Graphical user interface, application

Description automatically generated]()

Figure 02 – Sign-Up Page

![Graphical user interface, application

Description automatically generated]()![Graphical user interface, text, website

Description automatically generated]()

Figure 03 – Homepage Figure 04 – Medicine Details

If the user is not a first-time user, the user should be able to see the home page after he log-in, see Figure 04. The home page contains 3 sections, 1st section is listing all the available medicine, 2nd section is “About us”, and the 3rd section is the cart button to let the customer/patient see his/her ordered products before proceeding to checkout.

# **4-Functional Requirements**

## 4.1 Customer class

### 4.1.1 Functional Requirements

**Title**: Customer signup

**Desc**: System shall allow customers to be able to sign up so they can be registered on the system.

Customer provides required info which includes:

* First Name
* Last Name
* Email (unrepeated email)
* Phone Number
* Password (at least 8 characters & contain a special character & at least one capital letter)
* Age (At least 16 years old)

### 4.1.2 Functional Requirements

**Title**: Customer login

**Desc**: System shall allow customers to be able to login to an existing account using their registered Phone number and Password.

### 4.1.3 Functional Requirements

**Title**: Delete Account

**Desc**: System shall allow customers to be able to Delete their account.

### 4.1.4 Functional Requirements

**Title**: Update Profile Info

**Desc**: System shall allow customers to be able to update their profile information.

Customer provides required info which includes:

* First Name
* Last Name
* Email (unrepeated email)
* Phone Number
* Password (at least 8 characters & contain a special character & at least one capital letter)
* Age (At least 16 years old)

### 4.1.5 Functional Requirements

**Title**: Add New Address

**Desc**: System shall allow customers to be able to add a new address to their addresses list.

Customer provides required info which includes:

* City
* Area
* St name
* Building number
* Floor number
* House number

### 4.1.6 Functional Requirements

**Title**: View Addresses

**Desc**: System shall allow customers to be able to view their addresses list.

### 4.1.7 Functional Requirements

**Title**: Payment Details

**Desc**: System shall allow customers to be able to add their payment cards.

User provides required info which includes:

* Card number (16 number)
* Exp date (month/year)
* CVV (3 numbers)

### 4.1.8 Functional Requirements

**Title**: View Visa

**Desc**: System shall display all the customer’s Visa.

### 4.1.9 Functional Requirements

**Title**: View Orders

**Desc**: System shall display all the orders that the customer made.

### 4.1.10 Functional Requirements

**Title**: Confirm Order

**Desc**: System shall allow customers to be able to proceed with confirming their order.

User provides required info which includes:

* Payment method
* Phone number & password (for confirmation)
* Chosen address (From the addresses list)
* Cart items

## 4.2 aDMIN class

### 4.2.1 Functional Requirements

**Title**: Add Employee

**Desc**: System shall allow Admin to be able to add a new employee.

Customer provides required info which includes:

* First Name
* Last Name
* Work Email (unrepeated email)
* Phone Number
* Password (at least 8 characters & contain a special character & at least one capital letter)
* Section
* ID

### 4.2.2 Functional Requirements

**Title**: Admin login

**Desc**: System shall allow Admin to be able to login to the system using their ID and Password.

### 4.2.3 Functional Requirements

**Title**: view customers

**Desc**: System shall display all the customers’ main details to the admin.

### 4.2.4 Functional Requirements

**Title**: View Full Customer Info

**Desc**: System shall display the customer information in details to the admin by providing the customer’s phone number.

### 4.2.5 Functional Requirements

**Title**: Delete Customer

**Desc**: System shall allow Admin to be able to delete any customer by using their phone number.

## 4.3 aDMIN class

### 4.3.1 Functional Requirements

**Title**: Add Product

**Desc**: System shall allow Admin to be able to add new medicine and products to the stock.

Admin provides required info which includes:

* Medicine name
* Company
* Active ingredients
* Price
* Amount
* photo

### 4.3.2 Functional Requirements

**Title**: top ten products

**Desc**: System shall allow customers to be able to view top 10 products.

### 4.3.3 Functional Requirements

**Title**: All products

**Desc**: System shall allow customers to be able to view all products.

### 4.3.4 Functional Requirements

**Title**: Category products

**Desc**: System shall allow customers to be able to view products according to their category.

## 4.4 Cart class

### 4.4.1 Functional Requirements

**Title**: Add to cart

**Desc**: System shall allow customer to be able to add medicine and products to cart.

### 4.4.2 Functional Requirements

**Title**: view cart

**Desc**: System shall allow customer to be able to view their cart products.

### 4.4.3 Functional Requirements

**Title**: delete from cart

**Desc**: System shall allow customer to be able to delete from their cart.

# **5-Non-Functional requirement**

## 5.1 Reliability

* We are expecting that the average time between failures must be 4 times maximum per year.

## 5.2 Recoverability

* It is expected that the System must recover after a breakdown within 2 hours.

## 5.3 Performance

* Expected time of loading pages in AMMY app should not exceed 1.5 second.

## 5.4 Availability

* The system should be available to use 24 hours & 7 days.

## 5.5 Usability

* We are expecting that the user would take at max 30 minutes to use at least 90% of the system.
* We are expecting that the admin would take 1 hour training time to manage the system.

## 5.6 Maintainability

* The system is using OOP, so any additional feature or operations can be added to the system afterwards.

## 5.7 Security

Data security should be maintained in our program to protect user information from hacking attempts and data corruption. We shall employ the following approaches to ensure data security, privacy, and system reliability:

* allowing only both read and write access to the database for system developers. Table structures cannot be deleted or changed by developers.

# **6-Diagrams**

## 6.1 Use case diagram

![Diagram

Description automatically generated]()

### 6.1.1 Use case scenarios

Use Case 01

|  |  |
| --- | --- |
|  | |
| Use case Name | Sign-up |
| Actors | Customer (First use) |
| Main success scenario: | 1. System prompts the Customer to login or signup with a new account.  2. Customer chooses signup  3.System prompts the Customer to enter the needed information first name, last name, phone number, etc.  4.The Customer fill in the information  5. Customer presses done  6. System checks if the phone number and email are unique.  7.system register the Customer as a new user. |
| Exceptions | 5a. The Customer presses done without filling all the information  6a. The customer enters a Invalid or used phone number and email |
| Actions | 5.1 System display error alert “Please Fill in all the needed information”.  5.2 The Customer completes the missing information.  6.1 The System display error alert “Invalid Phone number or email”.  6.2 The Customer enter a different phone number or email. |
| Pre-condition | The Customer opens the app for the first time |
| Post condition | The Customer is registered successfully.  The Customer information is stored in our database. |

### 6.1.2 Use case scenarios

Use Case 02

|  |  |
| --- | --- |
|  | |
| Use case Name | Add employee |
| Actors | Admin(Already Existing in the system) |
| Main success scenario: | 1.The Admin choses “Add employee” option.  2.System prompts the Admin to enter the needed information first name, last name, Work email, ID, etc.  4.The Admin fills in the information.  5. Admin presses done.  6. System checks if the ID and work email are unique.  7.system register the new employee. |
| Exceptions | 5a. The Admin presses done without filling all the information  6a. The Admin enters a Invalid or existing ID and Work email. |
| Actions | 5.1 System displays error alert “Please Fill in all the needed information”.  5.2 The Admin completes the missing information.  6.1 The System displays error alert “Invalid ID or Work email”.  6.2 The Admin enter a different ID or Work email. |
| Pre-condition | An existing admin in the system is logged in. |
| Post condition | The New employee is registered successfully.  The New employee information is stored in our database.  The New employee can now login to the system. |

### 6.1.3 Use case scenarios

Use Case 03

|  |  |
| --- | --- |
|  | |
| Use case Name | Login |
| Actors | Admin/Customer |
| Main success scenario: | 1.The Admin/customer open the app and chooses to login.  2.System prompts the Admin/Customer to enter the needed information to login ID/Phone number and password.  4.The Admin/Customer enter the information.  5. Admin/Customer presses done.  6. System checks if the ID/Phone number and password exists in the database.  7.system allows Admin/Customer into the system. |
| Exceptions | 5a. The Admin/Customer presses done without filling in all the information.  6a. The Admin/Customer enters an incorrect ID/Phone number and password. |
| Actions | 5.1 System displays error alert “Please Fill in all the missing columns”.  5.2 The Admin/Customer completes the missing information.  6.1 The System displays error alert “Invalid Values entered”.  6.2 The Admin/Customer enter a the correct ID/Phone number or password. |
| Pre-condition | The Admin/Customer already exists in the Database. |
| Post condition | The Admin /Customer is logged in successfully. |

### 6.1.4 Use case scenarios

Use Case 04

|  |  |
| --- | --- |
|  | |
| Use case Name | Delete customer |
| Actors | Admin/Customer |
| Main success scenario: | 1.The Admin/customer open the app and chooses to Delete Customer.  2.System prompts the Admin/Customer to enter the needed information to delete the account Phone number and password.  4.The Admin/Customer enter the information.  5. Admin/Customer presses done.  6. System checks if the Phone number and password exists in the database.  7.system deletes the account successfully. |
| Exceptions | 5a. The Admin/Customer presses done without filling in all the information.  6a. The Admin/Customer enters an incorrect Phone number and password. |
| Actions | 5.1 System displays error alert “Please Fill in all the missing columns”.  5.2 The Admin/Customer completes the missing information.  6.1 The System displays error alert “Invalid Values entered”.  6.2 The Admin/Customer enter the correct Phone number and password. |
| Pre-condition | The Customer who is going to be deleted already exists in the Database. |
| Post condition | The deleted Customer is removed successfully. |

### 6.1.5 Use case scenarios

Use Case 05

|  |  |
| --- | --- |
|  | |
| Use case Name | View all products |
| Actors | Customer |
| Main success scenario: | 1.The customer chooses to view all products.  2.System displays all the products in the system. |
| Post condition | System displays all the results found. |

### 6.1.6 Use case scenarios

Use Case 06

|  |  |
| --- | --- |
|  | |
| Use case Name | View customers |
| Actors | Admin |
| Main success scenario: | 1.The Admin chooses “View customers” option.  2.System retrieves the Customers list from the Database.  3.The system displays the Customers list to the admin. |
| Pre-condition | The Admin is Logged in to the system. |
| Post condition | The system displays the customers list sussesfully. |

### 6.1.7 Use case scenarios

Use Case 07

|  |  |
| --- | --- |
|  | |
| Use case Name | Add to cart |
| Actors | Customer |
| Main success scenario: | 1.The customer opens the page of a medicine and press “add to cart”.  2.The system adds the product to the customer’s cart list.  3.the system saves the list in the database. |
| Exceptions | 2a.The product is out of stock. |
| Actions | 2.1 System displays error alert “Sorry,The product is out of stock right now”. |
| Pre-condition | Customer viewed the product page. |
| Post condition | The system adds the product to the cart list and save it to the database. |

### 6.1.8 Use case scenarios

Use Case 08

|  |  |
| --- | --- |
|  | |
| Use case Name | View orders |
| Actors | Customer |
| Main success scenario: | 1.The Customer chooses “View orders” option.  2.System retrieves the Customer’s history orders list from the Database.  3.The system displays the orders list to the Customer. |
| Exceptions | 2a. The Customer did not make any order before. |
| Actions | 5.1 System displays error alert “No Orders found”. |
| Pre-condition | The Customer made at least 1 order before. |
| Post condition | The system displays the order list to the customer. |

### 6.1.9 Use case scenarios

Use Case 09

|  |  |
| --- | --- |
|  | |
| Use case Name | confirm order |
| Actors | Customer |
| Main success scenario: | 1.The customer chooses to confirm their order.  2.The System prompts the Customer to choose a payment card.  3.The customer chooses a card from his card lists or adds a new one.  4.The system checks if the card is valid.  5.The system prompts the Customer to choose which address to use.  6.The customer choose an address from the addresses list.  7.The system adds the medicine in the cart list and ask the customer to enter their Phone number and password for confirmation.  8.The Customer enter their Phone number and password then press confirm.  9.The system confirms the order and take the medicine list from the database. |
| Exceptions | 4a. The card is not valid.  8a. The Customer enters an incorrect Phone number and password. |
| Actions | 4.1 System displays error alert “The card is not valid”.  4.2 The Customer enter a new valid card.  8.1 The System displays error alert “Invalid Values entered”.  8.2 The Customer enter the correct Phone number and password. |
| Pre-condition | The Customer have at least one item in the cart list.  The customer have at least one address registered. |
| Post condition | The system adds the order to the orders list of the customer in the database.  The system prints the customer’s order receipt. |

### 6.1.10 Use case scenarios

Use Case 10

|  |  |
| --- | --- |
|  | |
| Use case Name | view stock |
| Actors | Admin |
| Main success scenario: | 1.The Admin chooses “view stock” option.  2.System retrieves all items in the stock from the Database.  3.The system displays the items list to the admin. |
| Pre-condition | The Admin is Logged in to the system. |
| Post condition | The system displays the items list successfully. |

### 6.1.11 Use case scenarios

Use Case 11

|  |  |
| --- | --- |
|  | |
| Use case Name | Add new address |
| Actors | Customer |
| Main success scenario: | 1.The Customer chooses “Add New Address” option.  2.System prompts the Customer to enter the needed information city, area, st name, etc.  3.The Customer fills in the information.  4.The Customer presses done.  5.The system adds the new address to the addresses list of the customer. |
| Exceptions | 4a. The Customer presses done without filling in all the information. |
| Actions | 5.1 System displays error alert “Please Fill in all the missing columns”.  4.2 The Customer completes the missing information. |
| Pre-condition | The Customer is already logged in the system. |
| Post condition | The system adds the address to the customer addresses list in the database. |

### 6.1.12 Use case scenarios

Use Case 12

|  |  |
| --- | --- |
|  | |
| Use case Name | Add product |
| Actors | Admin |
| Main success scenario: | 1.The Admin chooses “Add product” option.  2.System prompts the Admin to enter the needed information name, company, active ingredient, etc.  3.The Admin fills in the information.  4.The Admin presses done.  5.The system adds the product to the database. |
| Exceptions | 4a. The Admin presses done without filling in all the information. |
| Actions | 5.1 System displays error alert “Please Fill in all the missing columns”.  4.2 The Admin completes the missing information. |
| Pre-condition | The Admin is already logged in the system. |
| Post condition | The system adds the product to the stock in the database. |

# 6.2-Sequence diagrams

## 6.2.1-ADD Visa

![Diagram

Description automatically generated]()

## 6.2.2-Confirm Order

![Diagram

Description automatically generated]()

## 7.3Class Diagram

Diagram

Description automatically generated

## 8.4 ER DIGRAM

Diagram

Description automatically generated

## 8.5-Relational Schema

![Diagram

Description automatically generated]()

### 8.5.1-Business Rules

1)Each Customer has a unique ID, First Name, Last Name, Email, Password, Age, Phone Number

2)Each Customer Address has City, Area, Street Name, Building Number, Floor, House Number

3)Each Customer Address must be linked with one and only customer

4)Each Customer may have one or more address or none

5)Each Order has a unique ID, Address, Phone Number, Payment Method, Visa, Total

6)Each Customer Order must be linked with at one and only customer

7)Each Customer may or may not have an order or multiple customer orders

8)Each Cart Item has Product, Quantity

9)Each Cart Item must be linked with one and only customer

10)Each Customer must have at least one cart item

11)Each Visa has a unique Card Number, CVV, EXP Date

12)Each Visa Card must be linked with one and only customer

13)Each Customer may have one or more Visa Card

14)Each Product has a unique ID, Name, Company, Price, Amount, Photo, and Active Ingredient can have multiple values

15)Each Product in cart can be in one or more cart

16)Each Cart must have one and only Product

17)A Product Description has Indications and Use, Dose, Precautions

18)Each Product must have one and only description

19)Each Product Description must be linked with one and only Product, otherwise, it will not exist

20)Each Delivery Order has Location, and Date

21)Each Order is delivered by one or more delivery man

22)Each Delivery employee can deliver one or more order

23)Each Employee has a unique Phone Number, a unique Employee ID, a unique Work Email, Salary, Name that contains his/her First Name and Last Name, and Department that can be APP Admin, Pharmacist, Delivery, or Cashier

24)Each APP Admin has an Admin Password

25)Each Pharmacist has Job registration number

26)Each Delivery has Orders Delivered

27)Each Cashier has Cashier Password

# 9-Test Cases

## 9.1-Test Case 01

|  |  |
| --- | --- |
| **Test Case ID: T\_C\_Signup** | **Test Designed by: Yousef Adel Khalil** |
| **Test Priority (Low/Medium/High):Medium** | **Test Designed date: 12/24/2022** |
| **Module Name: Customer Signup** | **Test Executed by: Mostafa Samer Dorrah** |
| **Test Title: Customer Signup with valid inputs** | **Test Execution date:12/24/2022** |
| **Description: The Customer should enter valid including correct form for the email and password.** |  |

|  |
| --- |
| **Pre-conditions: The Customer just opened our website and doesn’t have an account** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** |
| **1-** | **Enter the First Name** | **First name: Walid** | **Customer should be Signed up and Enter the home page** | **The Customer successfully entered the system and is in the homepage** | **Pass** |
| **2-** | **Enters the Last Name** | **Last Name: Ramy** |  |  |  |
| **3-** | **Enter The Unique Phone Number** | **Phone Number:1055906711** |  |  |  |
| **4-** | **Enter a valid email** | **Email:WalidRamy@gmail.com** |  |  |  |
| **5-** | **Enter a valid Password** | **Walid@28Ramy** |  |  |  |
| **6-** | **Enter Age** | **28** |  |  |  |
| **7-** | **Press”Dones”** |  |  |  |  |

|  |
| --- |
| **Post-conditions: The Customer data is saved in the database and is given a unique ID** |

## 9.2-Test Case02

|  |  |
| --- | --- |
| **Test Case ID:T\_C\_ADDCart** | **Test Designed by: Mostafa Samer Dorrah** |
| **Test Priority (Low/Medium/High):High** | **Test Designed date:12/24/2022** |
| **Module Name: Customer Cart** | **Test Executed by: Yousef Adel Khalil** |
| **Test Title: Customer Add to cart** | **Test Execution date: 12/24/2022** |
| **Description: The Customer try to add a product to their cart** |  |

|  |
| --- |
| **Pre-conditions: The Customer is Logged into the website. The Customer is in the product page** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** |
| **1-** | **Choose the needed amount** | **Amount: 4** | **The Product should be added to the cart with quantity of 4** | **The Product is added to the Customer’s cart and the Quantity next to the product is 4** | **Pass** |
| **2-** | **Press” ADD”** |  |  |  |  |

|  |
| --- |
| **Post-conditions: The Product is saved in the Database Table “Cart” under the ID of the customer with the chosen quantity.** |

## 9.3-Test Case03

|  |  |
| --- | --- |
| **Test Case ID:T\_C\_UpdateCart** | **Test Designed by: Abdelrahman Mohamed Elatrozy** |
| **Test Priority (Low/Medium/High):High** | **Test Designed date:12/24/2022** |
| **Module Name: Customer Cart** | **Test Executed by: Mohamed Ahmed Mohamed** |
| **Test Title: Customer Update amount in cart** | **Test Execution date: 12/24/2022** |
| **Description: The Customer try to change the product amount in their cart** |  |

|  |
| --- |
| **Pre-conditions: The Customer is Logged into the website. The Customer is in the Cart page** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** |
| **1-** | **Next to the product press “+” to add one more** | **Quantity: 5** | **The Quantity next to the product should be 5** | **The Page refreshed and the Quantity is now 5** | **Pass** |
| **2-** | **Press” Done”** |  |  |  |  |

|  |
| --- |
| **Post-conditions: The Product Quantity is updated in the Database with the chosen new quantity.** |

## 9.4-Test Case04

|  |  |
| --- | --- |
| **Test Case ID:T\_C\_ViewOrders** | **Test Designed by: Mohamed Ahmed Mohamed** |
| **Test Priority (Low/Medium/High): Medium** | **Test Designed date:12/24/2022** |
| **Module Name: Customer orders** | **Test Executed by: Abdelrahman Mohamed Elatrozy** |
| **Test Title: Customer View Orders** | **Test Execution date: 12/24/2022** |
| **Description: The Customer views the past orders** |  |

|  |
| --- |
| **Pre-conditions: The Customer have complected at least one order** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** |
| **1-** | **Press”View orders”** |  | **The system should Display all orders.** | **The system Displays all the customer orders on the screen.** | **Pass** |

## 9.2-Test Case05

|  |  |
| --- | --- |
| **Test Case ID:NT\_A\_Login** | **Test Designed by: Mostafa Samer Dorrah** |
| **Test Priority (Low/Medium/High): High** | **Test Designed date:12/23/2022** |
| **Module Name: Admin login** | **Test Executed by: Mostafa Samer Dorrah** |
| **Test Title: Admin incorrect login** | **Test Execution date: 12/24/2022** |
| **Description: The Admin enter invalid login info(Negative Test)** |  |

|  |
| --- |
| **Pre-conditions: The Admin already have an app admin account.** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| **1-** | **Enter invalid Work Email** | **Work Email: a\_mostfa@ammy.com** | **The system will not allow the Admin to login** | **The system did not allow the Admin to login** | **Pass** | **The correct Email is: a\_mostafa@ammy.com** |
| **2-** | **Enter Invalid Password** | **Password: Mo@11** |  |  |  | **The correct password is:** **Mo@101** |
| **3-** | **Press”login”** |  |  |  |  |  |